The GAN that Warped: Semantic Attribute Editing with Unpaired Data

Garoe Dorta [1,2] Sara Vicente [2] Neill D. F. Campbell [1] Ivor J. A. Simpson [2,3]
[1]University of Bath [2]Anthropics Technology Ltd. [3]University of Sussex
{g.dorta.perez,n.campbell}@bath.ac.uk sara@anthropics.com i.simpson@sussex.ac.uk

(a) Input image (previously unseen) (b) User requested edit: "beak larger than head" (c) User requested edit: "beak smaller than head"

Figure 1: Semantic image editing at high resolution (2480×1850). The user requests a change in a semantic attribute and the input image (a) is automatically transformed by our method into, *e.g*let@tokeneonedot, an image with "beak larger than head" (b) or "beak smaller than head" (c). The content of the original input, including fine details, is preserved. Our focus is on face editing, as previous work, yet the method is general enough to be applied to other datasets. Please see the supplemental material for videos of these and other edits. (Zoom in for details)[1]

Abstract

Deep neural networks have recently been used to edit images with great success, in particular for faces. However, they are often limited to only being able to work at a restricted range of resolutions. Many methods are so flexible that face edits can often result in an unwanted loss of identity. This work proposes to learn how to perform semantic image edits through the application of smooth warp fields. Previous approaches that attempted to use warping for semantic edits required paired data, *i.e*let@tokeneonedotexample images of the same subject with different semantic attributes. In contrast, we employ recent advances in Generative Adversarial Networks that allow our model to be trained with unpaired data. We demonstrate face editing at very high resolutions (4k images) with a single forward pass of a deep network at a lower resolution. We also show that our edits are substantially better at preserving the subject's identity. The robustness of our approach is demonstrated by showing plausible image editing results on the Cub200 [34] birds dataset. To our knowledge this has not been previously accomplished, due the challenging nature of the dataset.

1 Introduction

Face editing has a long history in computer vision [22, 25, 32] and has been made increasingly relevant with the rise in the number of pictures people take of themselves or others. The type of edits that are performed usually manipulate a semantic attribute, such as removing a moustache or changing the subject's expression from a frown

[1]Image courtesy of Flickr user Christoph Landers.

to a smile.

In the last few years, deep learning approaches have become the standard in most editing tasks, including inpainting [26] and super-resolution [19]. Particularly, image-to-image translation methods [15] have been proposed, which learn how to transform an image from a source domain to a target domain. Cycle-GAN [41] allows learning such translations from unpaired data, *i.e*let@tokeneonedot for each source image in the dataset a corresponding target image is not required.

We are interested in photo-realistic image editing, which is a subset of image-to-image translation. We also focus on methods that provide a simple interface for users to edit images, *i.e*let@tokeneonedota single control per semantic attribute [6, 28], as this makes editing easier for novice users.

A disadvantage of current methods for editing [15, 6, 37] is that they focus on binary attribute changes. In order to allow partial edits an extensive collection of soft attribute data is usually required, which is labor intensive. Also, at inference each intermediate value requires a forward pass of the network, creating increased computational expense [28].

Most deep learning methods for image editing predict the pixel values of the resulting image directly [6, 7, 27, 28]. As a consequence these methods are only effective on images that have a similar resolution to the training data.

Recently, some interesting approaches that do allow edits at higher resolutions have been proposed. They proceed by estimating the edits at a fixed resolution and then applying them to images at a higher resolution. The types of possible edits are restricted to either warping [37] or local linear color transforms [11]. However, these approaches are limited by requiring paired data, *i.e*let@tokeneonedot for each source image in the dataset, they need the corresponding edited image.

Inspired by these high resolution methods, we introduce an approach to learn smooth warp fields for semantic image editing without the requirement of paired training data samples. This is achieved by exploiting the recent advances in learning edits from unpaired data with cycle-consistency checks, which derive from the Cycle-GAN [41] method. Our proposed model uses a similar framework to StarGAN [6] (an extension of Cycle-GAN) to predict warp fields that transform the image to achieve the requested edits. As the predicted warp fields are smooth, they can be trivially upsampled and applied at high resolutions.

A potential criticism is that there are clear limitations to the types of edits possible through warping. We argue that, for the changes that *can* be described in this way, there are several distinct benefits. The advantages of using warping with respect to pixel based models can be summarized as:

i. Smooth warp fields can be upsampled and applied to higher resolution images with a minimal loss of fidelity. This is opposed to upsampling images, which commonly results in unrealistic high frequency details. We show plausible edits using warp fields upscaled by up to 30$\times$ the resolution they were estimated at.

ii. Geometric transformations are a subset of image transformation models. These models make it easier to add priors to regularize against unrealistic edits. We demonstrate that editing by warping leads to a model that is better at preserving a subject's identity.

iii. Warp fields are more interpretable than pixelwise differences. We illustrate this with maps showing the degree of local stretching or squashing.

iv. Warp fields are much more suited to allow partial edits than pixel based approaches. We demonstrate the simplest implementation of this by scaling the warp field to show interpolation and extrapolation, and qualitatively show edits that are plausible.

An additional contribution of this work is to improve the specificity of editing attributes in StarGAN based models. We have observed that when these models are trained with several binary labels, they can transform more than one attribute of the image, even if only a single attribute should be edited. This is caused by the model having no indication of the attributes that should be edited, only of the final expected labels. For example, when enlarging the nose of a subject that has a slight smile, the model will not only make the nose bigger, but also make the smile more pronounced. In order to overcome this limitation, we propose to transform the labels to inform the model of which attributes should be edited, and which should remain fixed. This produces only the expected changes, and it does not require any extra label

Method	Unpaired data	High resolution	Forward pass
StarGAN [6]	✓		✓
FaceShop [27]	✓		✓
WG-GAN [10]			✓
FlowVAE [37]		✓	✓
CWF [9]		∼	✓
DBL [11]		✓	✓
iGAN [40]	✓	∼	
DFI [33]	✓	∼	
RelGAN [35]	✓		✓
SPM+R [36]	✓	∼	✓
Ours	✓	✓	✓

Table 1: Compared to previous work on image-to-image translation, our model is the only one that is able to edit high-resolution images in a single forward pass of the network, without paired training data. Partial fulfillment of the criterion is denoted by ∼.

annotation. Moreover, it removes the need to rely on a label classifier during inference.

We demonstrate the advantages of our contributions by providing quantitative and qualitative results by manipulating facial expressions and semantic attributes.

2 Previous work

This work builds upon recent work in image-to-image translation. These models can be used to modify the semantic attributes of an image. Our novelty is in describing these edits as smooth deformation fields, rather than producing an entirely new image. Smooth warp fields can be upsampled and applied to higher resolution images with a minimal loss of fidelity. Some previous works that allow high resolution editing rely upon paired data examples or require costly optimization, rather than a single forward pass of a network; neither of which is required for the proposed approach. An overview of the characteristics of our work compared to previous methods is shown in Table 1.

2.1 Image-to-Image translation

The Pix2Pix [15] model learns to transform an image from a source domain to a target domain using an adversarial loss [12]. This approach requires paired training data; *i.e*let@tokeneonedoteach image in the source domain must have a corresponding image in the target domain. Given this restriction, the method is often applied to problems where collecting paired data is easier, such as colorization.

Several extensions have been proposed to perform image-to-image translation without requiring paired data. In Cycle-GANs [41], two generators are trained, from the source to the target domain and vice versa, with a cycle-consistency loss on the generation process. However, this does not scale well with an increase in the number of domains, since 2 generators and 2 discriminators are needed for each domain pair. StarGAN [6] addresses this issue by conditioning the generator on a domain vector, and adding a domain classification output layer to the discriminator.

These models can find undesired correlations within a domain, which lead to changes in unexpected parts of the image. At least two techniques have been explored in order to encourage localized edits. Editing with residual images [30], and restricting the changes to a region given by a mask [24, 28]. The first is an overcomplicated representation for edits describing shape changes. It has to model the content in the region, subtract it, and add it in a second region. The second complicates the model significantly by adding an unsupervised mask prediction network.

Preceding this publication, two relevant extensions to StarGAN have been proposed: RelGAN [35] and SPM+R [36]. RelGAN proposes a binary label transformation approach similar to ours. However, their method is trained using a conditional adversarial loss that takes triplets composed of two images and a vector of changed attributes. In contrast, our approach uses a simpler classification loss, where only modified attributes count. RelGAN also enables partial editing, however it requires a forward pass of the network for each edit strength. In contrast, our approach trivially enables partial editing as a consequence of the edit being performed through warping. Similarly to our work, SPM+R suggests using a warping function to edit images; yet, this is followed by inpainting, which is not resolution agnostic. They do not demonstrate their approach for editing high resolution images ($>512\times512$), or on a more complex dataset such as Cub200. A further distinction is that instead of using a simple smoothness loss, as we propose, they use a warp field discriminator. Their resulting warp fields appear substantially less smooth and less sparse than the ones obtained by our approach.

2.2 Editing of high resolution images

Methods for editing images at high resolution can be divided into two categories: (i) those that use intermediate representations designed to upsample well, and (ii) those that directly predict pixel values at high resolutions.

Methods designed for upsampling These approaches are based on predicting constrained intermediate representations that are relatively agnostic to image resolution; *e.g*let@tokeneonedotwarp fields or local color affine transformations.

Warp fields, if sufficiently smooth, can be predicted at a lower resolution, upsampled and applied at high resolution with minimal loss of accuracy. Previous work has applied them to: redirecting eye gaze [9], editing emotional expressions [37] and synthesizing objects in novel views [39]. However, these methods require paired training data.

Spatial Transformer GANs [20] predict a global affine deformation for image compositing. Although the deformation can be applied at arbitrary resolutions, face editing by compositing is limiting, as it requires an infeasibly large dataset of suitable face parts to use as foreground images.

Local affine color transformations [5, 11] have been predicted from low resolution images and applied at the original resolution. However, these methods require paired data and have limited capacity for making semantic changes.

Blendshapes have been used as an intermediate representation to edit expressions in the context of video reenactment [31, 23]. Similar to our approach, the blendshape weights are resolution independent. However, several input video frames are required for the blendshape face model.

Rather than predict intermediate representations, iGAN [40] trains a low-resolution GAN and then fit a dense warp field and local affine color transformation to a pair of input-output images. The GAN generator is unaware of these restricted transformations, so it may learn edits that are not representable by such transformations.

Direct prediction at high resolution Several techniques have been proposed to scale deep image synthesis methods to larger resolutions. These include: synthesizing images in a pyramid of increasing resolutions [8], employing fully convolutional networks trained on patches [19], and directly in full resolution [3, 16]. These methods have been successfully applied for image enhancement [14] and face editing [27, 10]. A limitation for direct or pyramid based approaches is that they do not scale well with resolution, while training on patches assumes that global image information is not needed for the edit.

A method that modifies an image by following the gradient directions of a pretrained classification network, until it is classified as having the target attributes was proposed in [33]. However, this approach fails when the input resolution differs significantly from the training data.

In WG-GAN [10] the input image is warped based on a target image and two GAN generators are used thereafter to synthesize new content. Contrary to our method, WG-GAN requires paired data during training, cannot be applied at arbitrary resolutions, does not provide semantic controls and does not support partial edits.

3 Background

We start by reviewing GAN [12], Cycle-GAN [41] and StarGAN [6], as the latter is the basis for our model.

Generative Adversarial Network (GAN) [12] models consist of two parts, a generator and a discriminator. The generator produces samples that resemble the data distribution samples, and the discriminator classifies data samples as real or fake. The discriminator is trained with the real examples drawn from a training set and the fake examples as the output of the generator. The generator is trained to fool the discriminator into classifying generated samples as real. Formally, GANs are defined by a minimax game objective:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}} [\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}} [\log(1 - D(G(\mathbf{z})))], \quad (1)$$

where $\mathbf{x}$ is a sample from the dataset empirical distribution $p(\mathbf{x})$, $\mathbf{z}$ is a random variable drawn from an arbitrary distribution $p(\mathbf{z})$, G is the generator and D is the discriminator.

Given two data domains, A and B, Cycle-GAN [41] learns a pair of transformations $G : A \rightarrow B$ and $H : B \rightarrow A$. Unlike previous approaches, [15], this does not require paired samples from A and B, but instead utilizes a cycle consistency loss ($\|\mathbf{x}_a - H(G(\mathbf{x}_a))\|_1$, where $\mathbf{x}_a$ is

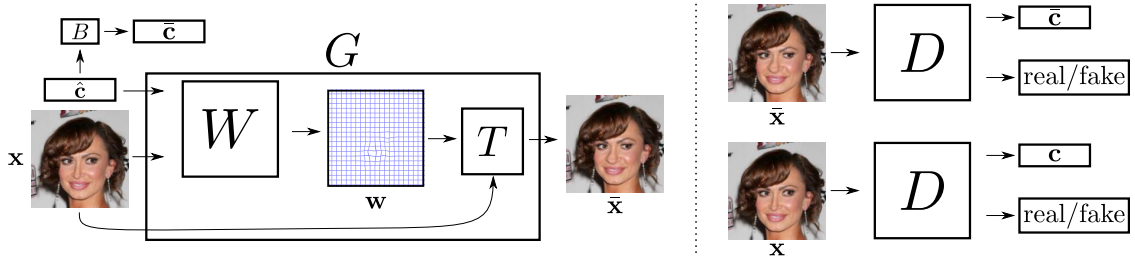

Figure 2: Overview of our model, which consists of a generator, G, and a discriminator, D. The generator contains a warping network, W, and a warping operator, T. The inputs to W are an RGB image, $\mathbf{x}$, and a transformed label vector, $\hat{\mathbf{c}}$. The output is a dense warp field, $\mathbf{w}$, which can be used by T to deform the input image and produce the output image, $\bar{\mathbf{x}}$. A label operator, B, converts the transformed labels, $\hat{\mathbf{c}}$, to binary labels, $\bar{\mathbf{c}}$. The discriminator evaluates both the input image, $\mathbf{x}$, and the generated image, $\bar{\mathbf{x}}$, for realism and the presence of attributes that agree with the labels. In this example, the only change between $\bar{\mathbf{c}}$ and $\mathbf{c}$ is the label for the attribute "big nose".

a sample image from domain A) to learn coherent transformations that preserve a reasonable amount of image content. An equivalent cycle loss is used for domain B. Cycle-GAN models are limited in that they require 2 generators and 2 discriminators for each domain pair.

Cycle-GAN was generalized by StarGAN [6] to require only a single generator and discriminator to translate between multiple domains. Here, each image $\mathbf{x}$ has a set of domains, represented as a binary vector $\mathbf{c}$. We use $(\mathbf{x}, \mathbf{c})$ to denote a pair sampled from the annotated data distribution. The generator, $G(\mathbf{x}, \bar{\mathbf{c}})$, transforms $\mathbf{x}$ to match the target domains indicated by $\bar{\mathbf{c}} \sim p(\mathbf{c})$, where $p(\mathbf{c})$ is the empirical domains distribution. The model is trained with:

i. a Wasserstein GAN [2] loss:

$$L_{adv} = \mathbb{E}_{\mathbf{x}}\left[D(\mathbf{x})\right] - \mathbb{E}_{\mathbf{x},\bar{\mathbf{c}}}\left[D(G(\mathbf{x}, \bar{\mathbf{c}}))\right], \quad (2)$$

ii. a Wasserstein gradient penalty [13] term:

$$L_{gp} = \mathbb{E}_{\dot{\mathbf{x}}}\left[\left(\|\nabla_{\dot{\mathbf{x}}} D(\dot{\mathbf{x}})\|_2 - 1\right)^2\right], \quad (3)$$

iii. a cycle consistency loss:

$$L_c = \mathbb{E}_{(\mathbf{x},\mathbf{c}),\bar{\mathbf{c}}}\left[\|\mathbf{x} - G(G(\mathbf{x}, \bar{\mathbf{c}}), \mathbf{c})\|_1\right], \quad (4)$$

iv. and domain classification losses:

$$L^d_{cls} = \mathbb{E}_{(\mathbf{x},\mathbf{c})}\left[-\log(C(\mathbf{x}, \mathbf{c}))\right] \quad (5)$$

$$L^g_{cls} = \mathbb{E}_{\mathbf{x},\bar{\mathbf{c}}}\left[-\log(C(G(\mathbf{x}, \bar{\mathbf{c}}), \bar{\mathbf{c}}))\right], \quad (6)$$

where $C(\mathbf{x}, \mathbf{c})$ is a classifier that outputs the probability that $\mathbf{x}$ has the associated domains $\mathbf{c}$, and $\dot{\mathbf{x}}$ is sampled uniformly along a line between a real and fake image. The classifier is trained on the training set (eq. 5) and eq. 6 ensures that the translated image matches the target domains.

4 Methodology

Our goal is to learn image transformations that can be applied at arbitrary scales without paired training data. An overview of our system is shown in Figure 2. We employ the StarGAN framework as the basis for our model and use the notation introduced above. As we focus on semantic face editing, we use indistinctly semantic attributes or binary labels to refer to the domains, $\mathbf{c}$ and $\bar{\mathbf{c}}$.

Warp parametrization We modify the generator such that the set of transformations is restricted to non-linear warps of the input image:

$$G(\mathbf{x}, \bar{\mathbf{c}}) = T(\mathbf{x}, W(\mathbf{x}, \bar{\mathbf{c}})), \quad (7)$$

where $W(\mathbf{x}, \bar{\mathbf{c}}) = \mathbf{w}$ is a function that generates the warp parameters. T is a predefined warping function that applies a warp to an image. W is chosen to be a neural network. We employ a dense warp parametrization, where $\mathbf{w}$ contains a displacement vector for each pixel in the input image. At train time, T warps the input according to the generated displacement field, $\mathbf{w}$, using bilinear interpolation. To improve image quality at inference time we use a bicubic interpolant.

4.1 Learning

We use the same adversarial losses (eq. 2 and eq. 3) and domain classification loss (eq. 5) as StarGAN.

Warp cycle loss The cycle consistency loss (eq. 4) is modified to produce warp fields that are inverse consistent, i.e. the composition of the forward and backward transformations yields an identity transformation:

$$L_c = \mathbb{E}_{(\mathbf{x},\mathbf{c}),\bar{\mathbf{c}}}\left[\|T(T(\mathbf{A},\mathbf{w}),\bar{\mathbf{w}})) - \mathbf{A}\|_2^2\right], \quad (8)$$

where $\bar{\mathbf{w}} = W(G(\mathbf{x},\bar{\mathbf{c}}),\mathbf{c})$, and $\mathbf{A}$ is a two channel image where each pixel takes the value of its coordinates. This loss is more informative than eq. 4, as a pixel loss provides no information for warps inside constant color regions.

Smoothness loss The warping network estimates an independent deformation per pixel. As such, there are no guarantees that the learned warps will be smooth. Therefore, an $L2$ penalty on the warp gradients is added to encourage smoothness. In practice a finite-difference approximation is used as

$$L_s = \mathbb{E}_{\mathbf{x},\bar{\mathbf{c}}}\left[\frac{1}{n}\sum_{(i,j)}\|\mathbf{w}_{i+1,j} - \mathbf{w}_{i,j}\|_2^2 + \|\mathbf{w}_{i,j+1} - \mathbf{w}_{i,j}\|_2^2\right], \quad (9)$$

where n is the number of pixels in the warp field, and $\mathbf{w}_{i,j}$ is the displacement vector at pixel coordinates (i,j).

Binary label transformation As mentioned in section 1, a StarGAN type model can make unexpected edits when modifying attributes. At inference time, the attribute classifier is used to infer the original labels. Depending on the desired edits, these labels are either changed or copied to the target vector. This means that the model cannot distinguish between the edited attributes and the copied ones. Thus, the model tends to accentuate the copied attributes.

To address this issue, we propose to explicitly instruct the generator on which attributes should be edited. The labels for the generator are transformed to contain three values, $[-1, 0, 1]$, where -1 indicates that the attribute should be reversed, 0 that it should remain unaffected, and 1 that it should be added. This approach has two distinct benefits. First, it leads to more localized edits. Second, it removes the need for a classifier during inference, as the unedited entries in the transformed target labels can be set to zero.

The classifier loss for the generator (eq. 6) is modified to only penalize the attributes that should be edited:

$$L_{cls}^g = \mathbb{E}_{\mathbf{x},\hat{\mathbf{c}}}\left[-h\sum_{i=0}^{r-1}|\hat{c}_i|\,\log\left(C(G(\mathbf{x},\hat{\mathbf{c}}),\bar{c}_i)\right)\right], \quad (10)$$

where $\hat{\mathbf{c}}$ are the transformed target labels, r is the number of attributes, and $h = r/\|\hat{\mathbf{c}}\|_1$ is a normalization factor, which ensures that there is no bias with respect to the number of edited attributes. During training, the transformed target label for each attribute, $\hat{c}_i$, is sampled independently from a Categorical distribution with probabilities $[0.25, 0.5, 0.25]$. As both types of labels are needed for the classification loss, a label operator, $\bar{c}_i = B(\hat{c}_i)$, is used to reverse the transformation, which is defined as $B(-1) = 0$ and $B(1) = 1$. $B(0)$ is undefined as its loss is zero by construction.

Complete objective The joint losses for the discriminator and the generator are defined as

$$L_D = -L_{adv} + \lambda_{gp}L_{gp} + \lambda_{cls}L_{cls}^d, \quad (11)$$

$$L_G = L_{adv} + \lambda_{cls}L_{cls}^g + \lambda_c L_c + \lambda_s L_s, \quad (12)$$

where λ_{cls}, λ_{gp}, λ_c and λ_s are hyper-parameters that control the relative strength of each loss. The classification loss in eq. 10 is used for images with several not mutually exclusive binary attributes, and eq. 6 is used otherwise.

4.2 Inference

Once the model parameters have been optimized, an input image of arbitrary size can be edited in a single forward pass of the network.

The input image is resized to match the resolution of the training data, and the transformed target labels, $\hat{\mathbf{c}}$, are set according to the desired edit. Then, the resized image and target labels are fed into the warping network, which produces a suitable warp field, $\mathbf{w}$. The warp field displacement vectors are rescaled and resampled to the original image resolution. Lastly, the original image is warped using the resampled warp field to produce the final edited image.

Figure 3: Employing a dense flow method [38] to transfer a "big nose" edit from StarGAN [6]. Results for the flow method with $\lambda = 0.05$ and $\lambda = 0.15$ are shown. StarGAN has edited the input to such lengths that good correspondences between the input and output cannot be found by the flow method.

5 Results

5.1 Datasets

We evaluate our method and baselines on a face dataset, CelebA [21] and a birds dataset, Cub200 [34].

CelebA The CelebA [21] dataset contains 202,599 images of faces and we use the train/test split recommended by the authors. Importantly, from the 40 binary attributes provided, we choose the ones more amenable to be characterized by warping, namely: smiling, big nose, arched eyebrows, narrow eyes and pointy nose.

Cub200 The Cub200 [34] dataset contains 11,788 images of 200 bird species. The images are annotated with 312 binary attributes and a semantic mask of the bird body. We choose the three binary attributes that correspond to the beak size relative to the head and remove the background using the semantic masks. The train/test split recommended by the authors is employed. Due to the alignment step discussed below, only 2,325 images are used for training.

Face alignment For both datasets, we make use of face landmark locations to align and resize the images to 128×128 using a global affine transformation, at both training and test time. At test time, the inverse of the affine transformation is used to transform the warp fields. The warp is then applied directly to the original image. This is in contrast with previous methods, that would edit the aligned image and then warp the edited image to the original space. For images outside of the test set, off-the-shelf methods [17] can be used to align them to the dataset.

5.2 Models

Our main baseline is StarGAN [6], a state of the art model for image-to-image translation. We define three novel models to evaluate our contributions. WarpGAN denotes models that output a warp field. A "+" suffix indicates models that employ our binary labels transformation scheme. Thus, StarGAN+ evaluates the effect of label transformation, and WarpGAN+ is our final proposed model.

An obvious alternative to our model consists of fitting a dense flow field to the results generated by StarGAN. We tested it using the dense optical flow matching technique described in [38], and we denote this method by SGFlow.

An example of SGFlow is shown in Fig. 3, using optical flow [38]. Warping based on optical flow may lead to artefacts when good correspondences are not found. Constraining StarGAN to generate images that are amenable to optical flow estimation is not trivial. Hence, this experiment shows that a naïve approach for applying the result of a StarGAN model to a higher resolution image is suboptimal. Thus, we drop SGFlow for the remaining experiments.

We also experimented with the GANimation [28] approach using the code provided by the authors. However we were unable to generate meaningful results when training the method with binary attributes. We suspect that this is due to the method's reliance on soft action unit labels.

Hyper-parameters All models were trained on a single Titan X GPU using TensorFlow [1]. The models hyper-parameters are: $\lambda_{\text{cls}} = 2$, $\lambda_{\text{gp}} = 10$, $\lambda_{\text{c}} = 10$ and $\lambda_{\text{s}} = 125$.

For the StarGAN baseline we employ the implementation provided by the authors, where we keep all their recommended hyper-parameters except for $\lambda_{\text{cls}} = 0.25$. The choice of λ_{cls}, for StarGAN and our models, is informed by the results shown in Fig. 8. Additional implementation details are provided in the supplementary material.

5.3 Qualitative results

We show qualitative results on the CelebA dataset in Fig. 4. For each input image, we show the edited images corresponding to changing a single attribute. StarGAN [6] often changes characteristics that are not related to the

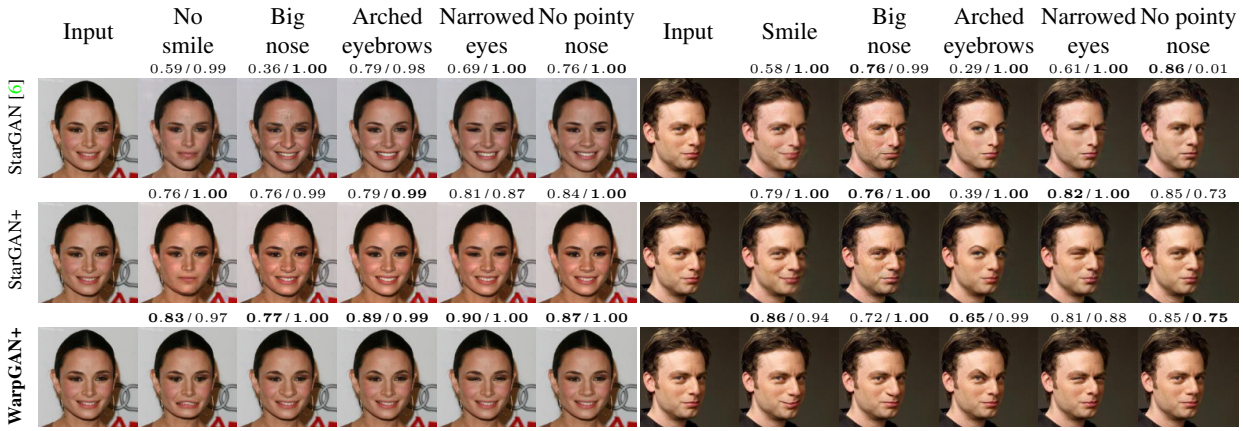

Figure 4: Comparison to previous work on the CelebA dataset. For a given input image, first column, each method attempts to transfer the semantic attribute in its corresponding column. A re-identification score and attribute probability are shown as (id / cls) on top of each image (higher is better). Our approach edits the attributes of the input images while better preserving the identity of the subject.

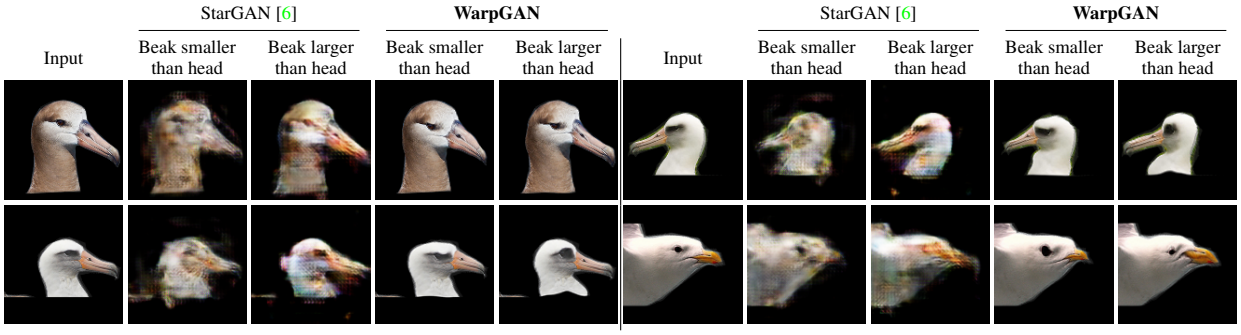

Figure 5: Comparison to previous work on the Cub200 dataset. Each model attempts to transfer the attribute (relative beak size) in each column to the input image. StarGAN is unable to produce good quality edits on this dataset, while our model edits the input images in a more plausible manner. Due to the more complex nature of this dataset, our model still struggles to produce artifact free transformations. Results from this model at the original image resolution, and without masking the background, can be found in the supplemental material.

perturbed attributes, such as the skin tone or the background color. StarGAN+ produces more localized edits than StarGAN. The WarpGAN+ edit for the "no smile" attribute is not particularly realistic. However, for most edits, our technique generates changes that are less exaggerated and better preserve the identity of the subject.

Qualitative results for the masked and aligned Cub200 dataset are shown in Fig. 5. Our approach is able to transfer the corresponding attribute, albeit sometimes producing unrealistic additional deformations. The poor quality results of StarGAN may be attributed to the increased complexity of this dataset and the reduced number of images, compared to CelebA. This is a generous comparison, as the predicted warp fields can be applied to the original images with complex backgrounds and at higher resolutions, as shown in the supplemental material. Fig. 1 demonstrates the power of the warping representation by operating at a far higher resolution (2480×1850) than can be achieved by direct methods.

Please see the supplemental material for animated edits and additional results, which also include more examples of failure cases.

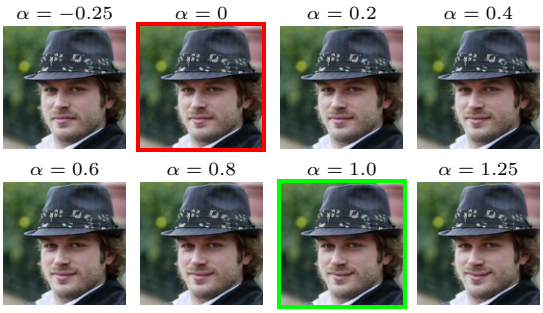

Figure 6: Partial editing with our model, for the "smile" attribute. A single warp is generated by our model, which is interpolated and extrapolated by scaling the magnitude of its values by α. The input image, $\alpha = 0$, is progressively edited in both directions.

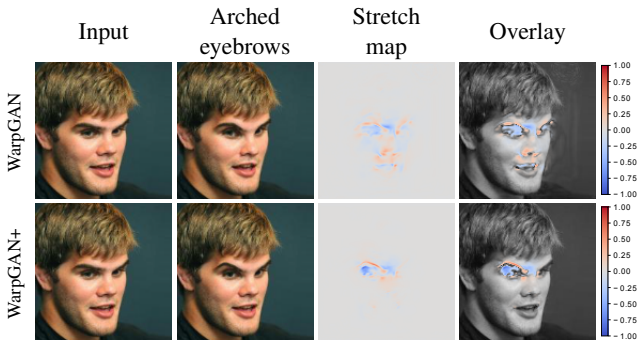

Figure 7: Stretch maps computed from the warp fields, for both WarpGAN and WarpGAN+. The log determinant of the Jacobian of the warp is shown, where blue indicates stretching and red corresponds to squashing. Our binary labels transformation scheme, used by WarpGAN+, leads to correctly localized edits.

Partial edits Another advantage of our model is that once a warp field has been computed for an input image, partial edits can be applied by simply scaling the predicted displacement vectors by a scalar, α. Results of interpolation and extrapolation of warp fields generated by our model are shown in Fig. 6. This is a cheap operation as it does not require to run the network for each new value of α, in contrast with previous methods that allow for partial edits [28]; this allows for edits to be performed at interactive speeds.

Visualizing warp fields A further advantage of our model is the interpretability of its edits. This is demonstrated in Fig. 7, where we show the log determinant of the Jacobian of the warp field, which illustrates image squashing and stretching. It can be seen how employing our binary label transformation scheme leads to more localized edits. Moreover, the values from the stretch maps can potentially be used to automatically determine which areas have been stretched or compressed excessively by the network. Thus they provide an intuitive measure to detect unrealistic edits.

5.4 Quantitative results

Quantitative evaluation is challenging for our setting. We provide two methodologies: the first measures the model performance based on separately trained networks, and the second is a user study to estimate perceptual quality.

Accuracy vs identity preservation We train a separate classifier on the training data, to estimate quantitatively if the edited images have the requested attributes. The classifier has the same architecture as the discriminator and is trained with the cross entropy loss of eq. 5. We also use a pretrained face re-identification model [29] to evaluate whether the edits preserve the identity.[2]

Results of both experiments are shown in Fig. 8, where an ideal editing model would be located on the top-right. On the x-axis we show the rate of images classified as having the target attribute (attribute accuracy), defined as $\frac{1}{m}\sum[C(\mathbf{x}, \bar{\mathbf{c}}) \geq 0.5]$, where m is the total number of images. On the y-axis, an identity preservation score is shown, which is evaluated as $1 - \frac{1}{m}\sum d(\mathbf{x}, \bar{\mathbf{x}})$, where $d(\cdot)$ is the L2 distance between the input and the edited image in the feature space of the face re-identification network. A distance larger than 1.2 (score lower than -0.2) has been used to indicate that two faces belong to different people [29].

There is a trade-off between attribute transfer and identity score. On one extreme, a new face that has the target attribute and does not match the original face would achieve maximal attribute accuracy with negative identity score. On the other, not modifying the image has maximal identity score, yet it does not achieve the target edit. With respect to StarGAN, our binary labels transformation scheme (StarGAN+) moves the curve towards higher attribute transfer with comparable identity score. Our warping approach (WarpGAN+) allows for stronger

[2] Additional details can be found in the supplemental material.

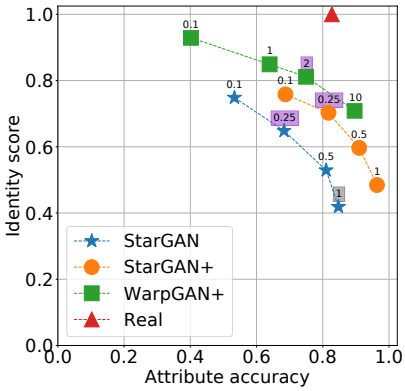

Figure 8: Presence of the edited attribute (x-axis) vs face re-identification score (y-axis), higher is better. The classification loss weight, λ_{cls}, is shown on top of each marker. Highlighted in gray is the value used by the StarGAN authors for this dataset, and in purple the ones used in this paper. Compared to previous work, our model produces edits that better preserve identity.

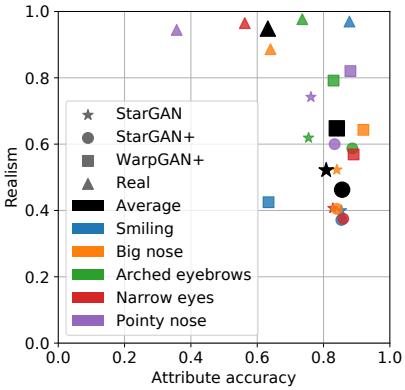

Figure 9: Human perception of the presence of the desired attribute (attribute accuracy) vs realism of the image, as indicated by the user study (higher is better). Images generated by our model are more realistic than those generated by previous work.

identity preservation than StarGAN+. Overall, our approach better preserves identity than previous work, for similar levels of attribute transfer. Moreover, we picked λ_{cls} based on these results: choosing the value that leads to both high accuracy and identity score.

Accuracy vs realism We perform a user study on Amazon Mechanical Turk (MTurk) to evaluate the quality of the generated images, for StarGAN, StarGAN+ and our model. For each method, we use the same 250 test images from CelebA and edit the same attribute per image.

We conducted two experiments, one to evaluate the realism of the images, where the workers had to answer whether the image presented was real or fake, and another to evaluate attribute editing, where we asked the workers whether the image contains the target attribute. In both experiments, the workers were randomly shown a single image at a time: either an edited image or an unaltered original image.[2]

Results of this user study are shown Fig. 9. A useful editing model has a high-level of realism and can produce the target edit. For the real data, the workers reliably evaluated image realism, however they were often inconsistent with the attribute labels. Nonetheless, the workers performance on real data should not be taken as an upper bound, as all methods tend to generate exaggerated edits to maximize correct classification. For the editing models, the attribute accuracy is consistent to that reported by the classifier network in Fig. 8. However, identity score and realism do not align, as they measure different notions. An image might contain only small edits, which the identity network is invariant to, yet those edits could include unrealistic artefacts that can be easily detected by humans. All editing models achieve good attribute transfer accuracy, with room for improvement mostly on the realism axis. Our model (WarpGAN+) achieves this for most attributes, and it is able to generate images that are more realistic than previous work.

6 Conclusions

This paper has introduced a novel way to learn how to perform semantic image edits from unpaired data using warp fields. We have demonstrated that, despite limitations on the set of edits that can be described using warping alone, there are clear advantages to modelling edits in this way: they better preserve the identity of the subject, they allow for partial edits, they are more interpretable, and they are applicable to arbitrary resolutions. Moreover, our binary label transformation scheme leads to increased performance, and removes the need to use a classifier during inference.

There are several avenues for future work, including different parametrizations for the warps, *e.g.*let@tokeneonedotin the form of velocity fields [4]. Additional intermediate representations that upsample

well could be added to increase the model flexibility, such as local color transformations [11]. Also, an inpainting method [26] could be locally applied in areas that have been warped or stretched excessively, which could be automatically detected using the log determinant of the Jacobian of the warp fields.

Acknowledgements

This work has been generously supported by Anthropics Technology Ltd., as well as the EPSRC CDE (EP/L016540/1) and CAMERA (EP/M023281/1) grants.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 7

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. 5

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 4

[4] Can Ceritoglu, Xiaoying Tang, Margaret Chow, Darian Hadjiabadi, Damish Shah, Timothy Brown, Muhammad H. Burhanullah, Huong Trinh, John T. Hsu, Katarina A. Ament, Deana Crocetti, Susumu Mori, Stewart H. Mostofsky, Steven Yantis, Michael I. Miller, and J. Tilak Ratnanather. Computational analysis of lddmm for brain mapping. *Frontiers in Neuroscience*, (7), 2013. 10

[5] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W. Hasinoff. Bilateral guided upsampling. *ACM Trans. Graph.*, 35(6):203:1–203:8, Nov. 2016. 4

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 4, 5, 7, 8

[7] Tali Dekel, Chuang Gan, Dilip Krishnan, Ce Liu, and William T. Freeman. Sparse, smart contours to represent and edit images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[8] Emily L Denton, Soumith Chintala, arthur szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc., 2015. 4

[9] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 311–326, Cham, 2016. Springer International Publishing. 3, 4

[10] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided gans for single-photo facial animation. In *SIGGRAPH Asia*. ACM, 2018. 3, 4

[11] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118, 2017. 2, 3, 4, 11

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 3, 4

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017. 5

[14] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. *arXiv preprint arXiv:1709.01118*, 2017. 4

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2, 3, 4

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 4

[17] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 7

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 27

[19] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew

Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 4

[20] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, December 2015. 7

[22] Zicheng Liu, Ying Shan, and Zhengyou Zhang. Expressive expression mapping with ratio images. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276. ACM, 2001. 1

[23] L. Ma and Z. Deng. Real-time facial expression transformation for monocular rgb video. *Computer Graphics Forum*, 2018. 4

[24] Youssef A Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image to image translation. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018. 3

[25] Umar Mohammed, Simon JD Prince, and Jan Kautz. Visio-lization: generating novel facial images. *ACM Transactions on Graphics (TOG)*, 28(3):57, 2009. 1

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 11

[27] Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Trans. Graph.*, 37(4):99:1–99:13, July 2018. 2, 3, 4

[28] A. Pumarola, A. Agudo, A.M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 7, 9

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 9

[30] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[31] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4

[32] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991. 1

[33] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep Feature Interpolation for image content changes. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 4

[34] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 1, 7

[35] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y. Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[36] Ruizheng Wu, Xin Tao, Xiaodong Gu, Xiaoyong Shen, and Jiaya Jia. Attribute-driven spontaneous motion in unpaired image translation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[37] Raymond A. Yeh, Ziwei Liu, Dan B Goldman, and Aseem Agarwala. Semantic facial expression editing using autoencoded flow. *arXiv preprint arXiv:1611.09961*, 2016. 2, 3, 4

[38] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition*, pages 214–223, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 7

[39] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In *Computer Vision – ECCV 2016*, pages 286–301. Springer International Publishing, 2016. 4

[40] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 597–613, Cham, 2016. Springer International Publishing. 3, 4

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 2, 3, 4

Appendix

A High-resolution Flickr faces

Figure 10: Additional results of our model on high-resolution images. Our model predicts warps at low resolution that can then be resized and applied to high resolution images. The model is able to keep the content and identity at high resolution. Please see supplemental videos demonstrating animated edits. Input images courtesy of Flickr users Kenneth DM and Randall Pugh. (Zoom in for details)

B High-resolution Flickr birds

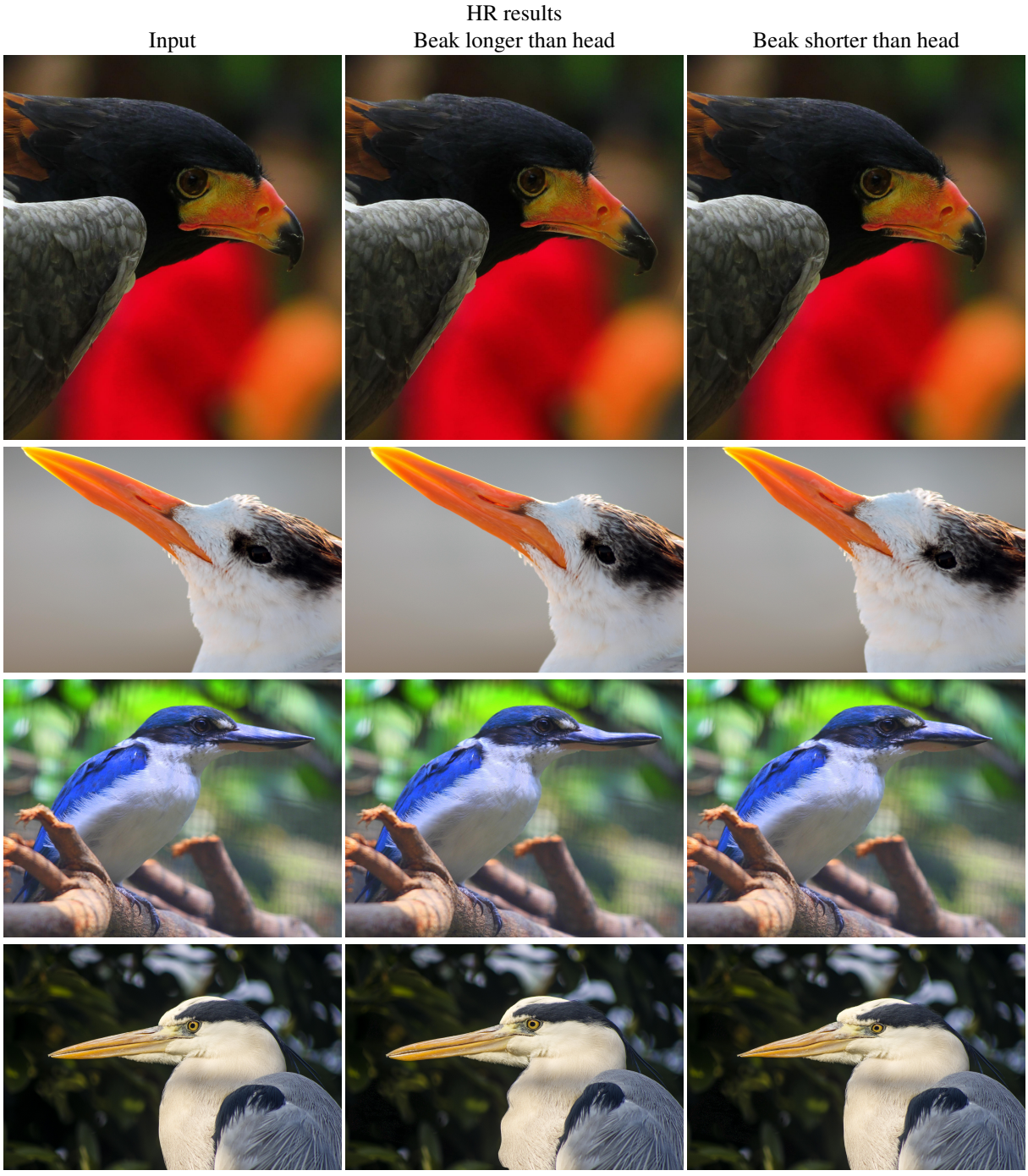

Figure 11: Additional results of our model on high-resolution images. Our model predicts warps at low resolution that can then be resized and applied to high resolution images. The model is able to keep the content and identity at high resolution. Please see supplemental videos demonstrating animated edits. Input images courtesy of Flickr users mickey, Lisa Leonardelli, and Andrey Grushnikov.

C Qualitative results on CelebA

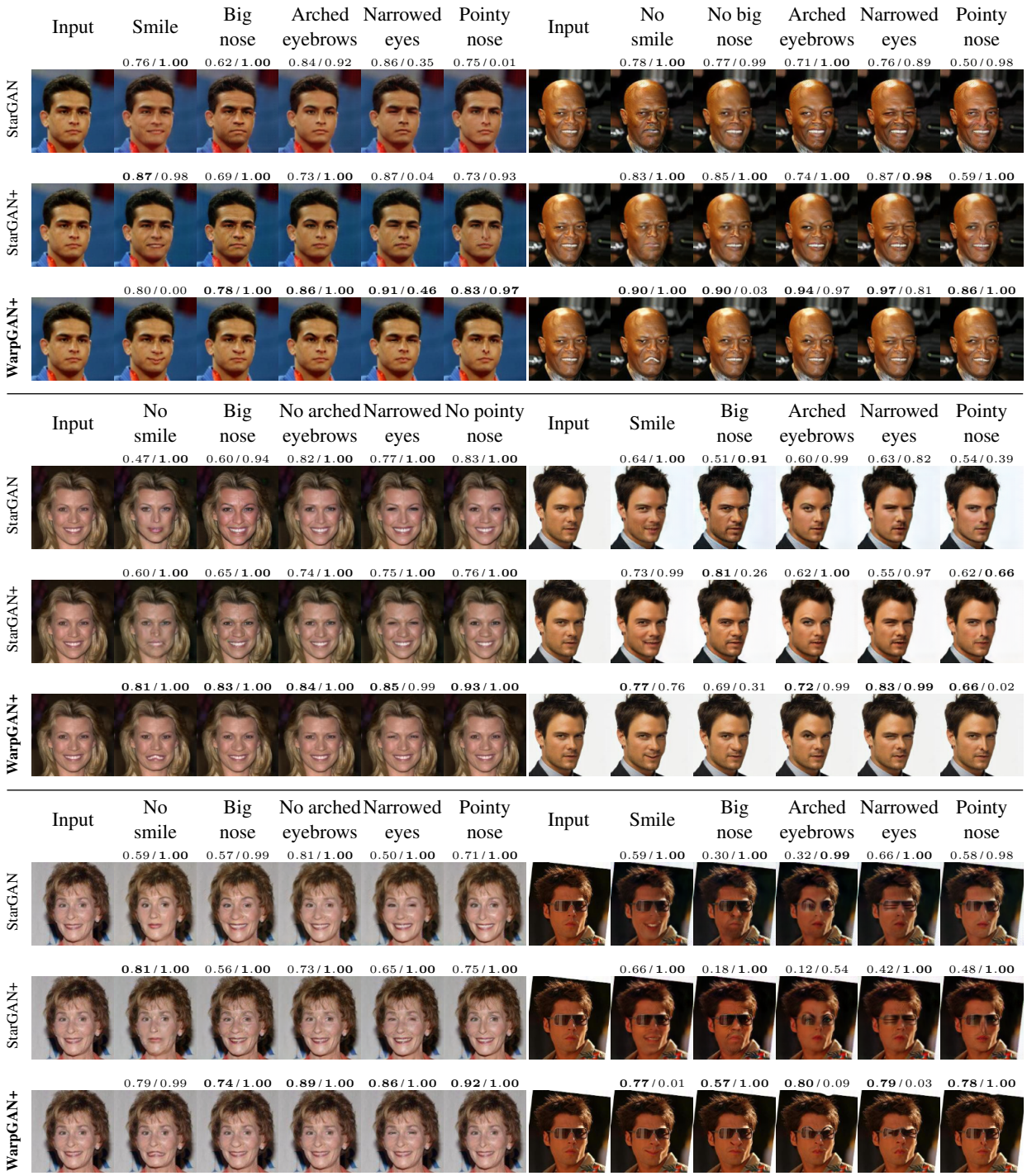

Figure 12: Comparison to previous work on the CelebA dataset. From a given input image, first column, each method attempts to transfer the semantic attribute in its corresponding column. On top of each image the re-identification score and the classification accuracy are shown as (id / cls) (higher is better). (Zoom in for details)

D Qualitative results on Cub200

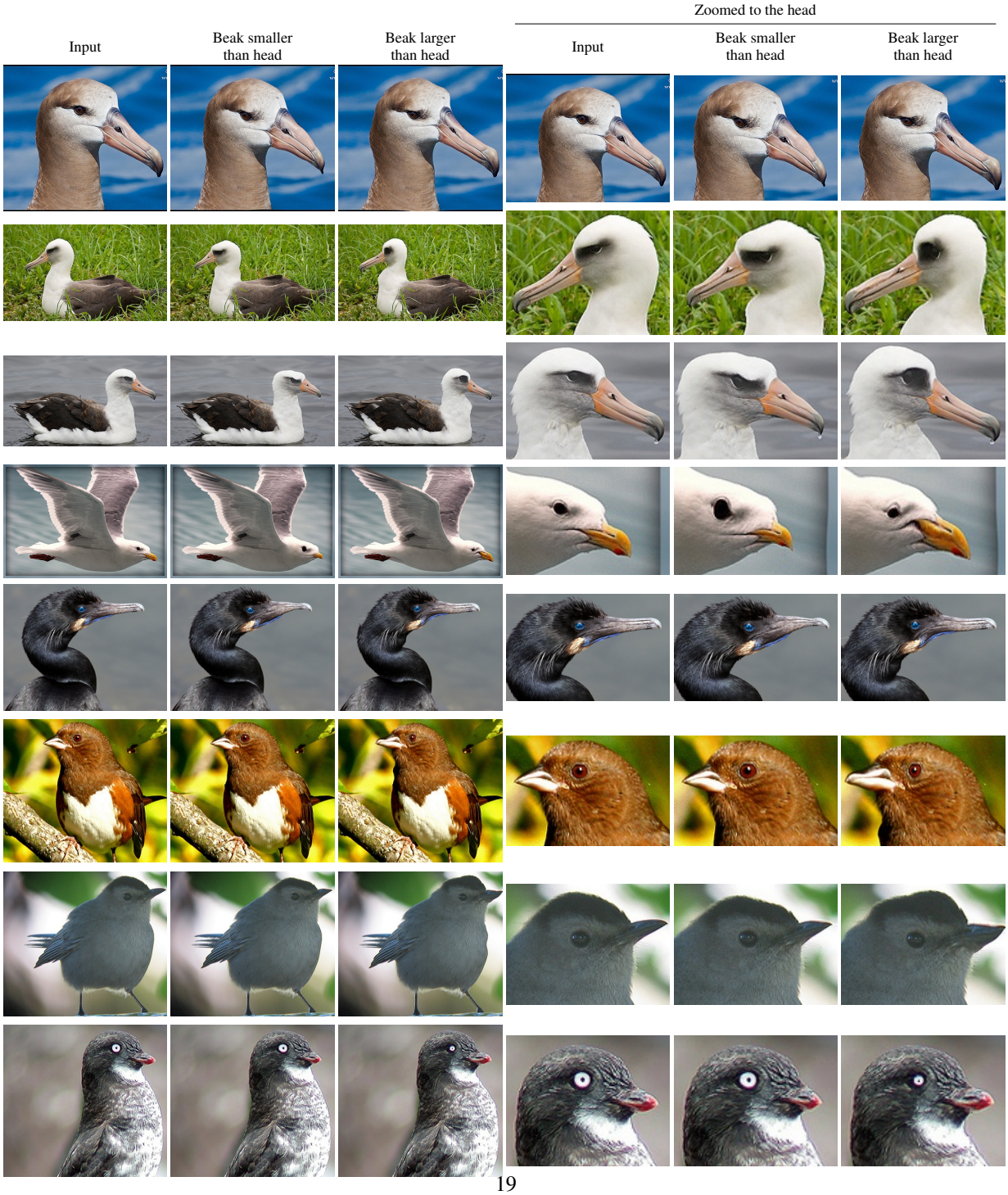

Figure 13: Additional results from our model on test images from the Cub200 dataset. The model attempts to transfer the attribute (relative beak size) in each column to the input image. For easiness of comparison, a crop of the head area is shown in the last three columns.

E Partial edits on CelebA

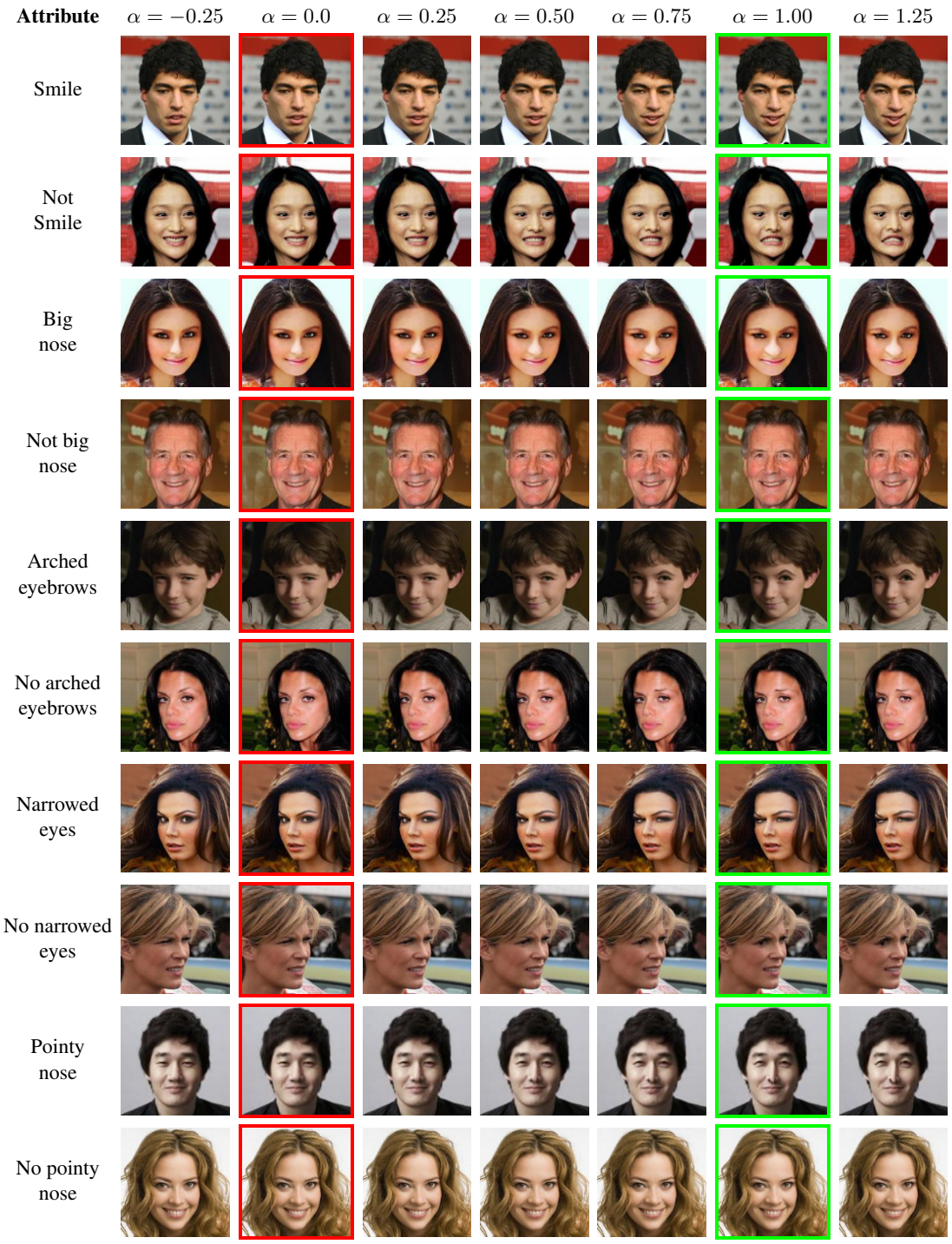

Figure 14: Partial editing with our model, for the the attribute indicated in the first column. A single warp is generated by our model, which is interpolated and extrapolated by scaling the magnitude of its values by α. The input image, $\alpha = 0$, is progressively edited in both directions. A red box denotes the input image, and a green one the output of the generator without α scaling. Please see supplemental videos demonstrating animated edits.

F Stretch maps on CelebA

Figure 15: Stretch maps computed from the warp fields, for WarpGAN and WarpGAN+. The log determinant of the Jacobian of the warp is shown, where blue indicates stretching and red corresponds to squashing. Our binary label transformation scheme (WarpGAN+) leads to correctly localized edits.

G Ablation study

G.1 Effect of each loss

In this section we evaluate the performance of the model after removing each of the losses, where "(w/o) Cycle" corresponds to removing L_{c}, "(w/o) Smooth" corresponds to removing L_{s}, "(w/o) Cls" corresponds to removing L_{cls}, "(pixel) Cycle" corresponds to using eq. 4 instead of eq. 8 in the paper, and "(w/o) Adv" corresponds to removing L_{adv} and L_{gp}.

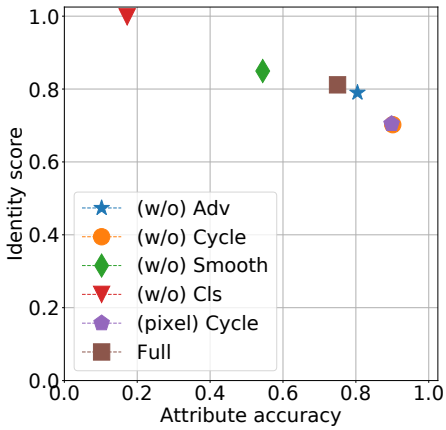

Figure 16: Presence of the edited attribute (x-axis) vs face re-identification score (y-axis), higher is better. Removing each loss in our model has a detrimental effect in either accuracy or identity preservation. The adversarial loss seems to have little effect, however, we qualitatively observed that without it, the edited images were less realistic.

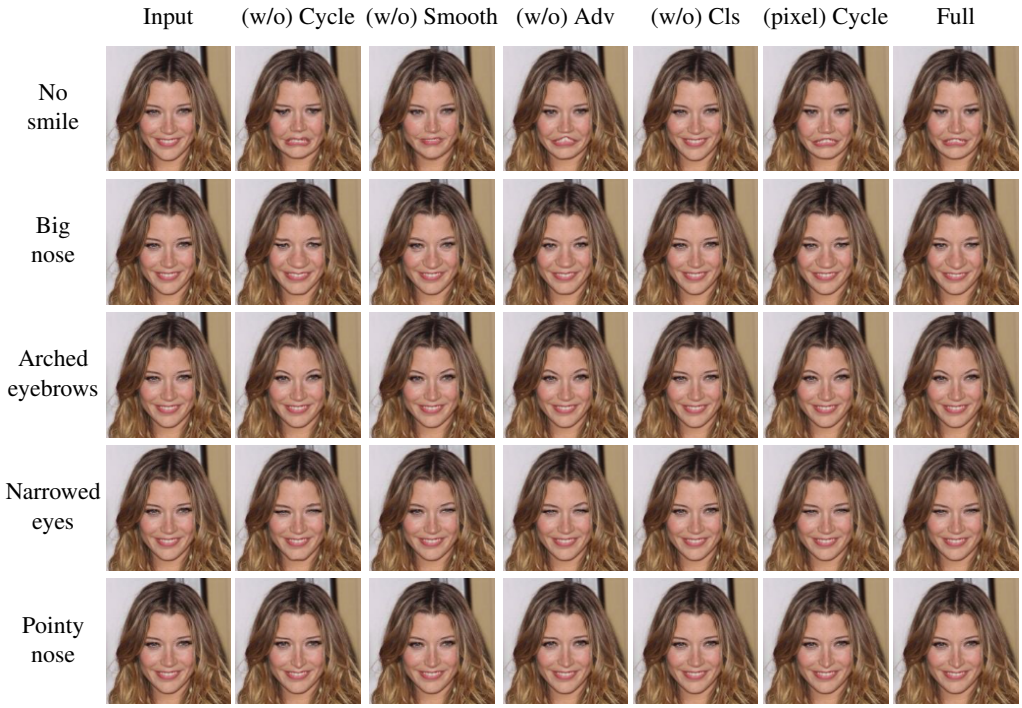

Figure 17: Ablation study, where we remove different losses in our model. For each loss, (w/o) Cycle: significant artifacts are introduced, (w/o) Smooth: leads to poor generalization, (w/o) Adv: unrealistic warps, (pixel) Cycle: exaggerated warps, and (w/o) Cls: trivial solution on the identity transform.

G.2 Effect of α

In this section we quantitatively evaluate the effect of scaling the displacement fields by a scalar α. For this experiment, we take WarpGAN+ trained with $\lambda_{\text{cls}} = 0.25$ and we evaluate the identity score and the attribute accuracy on the test set for different values of α. Results are shown in Fig. 18 for this model, which is denoted as WarpGAN+α. The curve produced by employing different values of α is very similar to the curve in Figure 8 in the paper, which was produced by modifying λ_{cls}. This implies that the model is relatively robust to the choice of λ_{cls}, as a similar effect to changing the value of λ_{cls} used during training can be achieved by choosing an alternative value of α at test time. This is in contrast to previous work, where modifying the strength of the effects requires training a model with the new parameters.

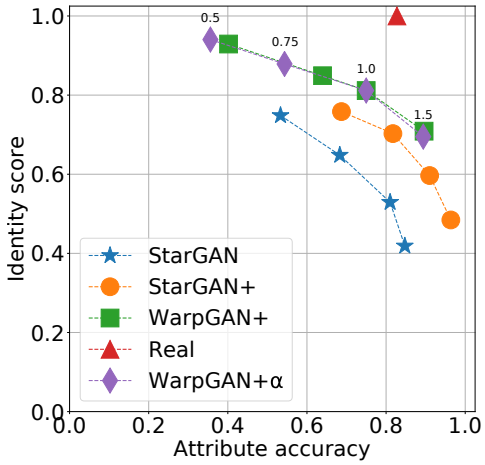

Figure 18: Presence of the edited attribute (x-axis) vs face re-identification score (y-axis), higher is better. For all models except WarpGAN+α, this figure is identical to Fig. 8 in the paper. For WarpGAN+α the value of α is shown on top of each marker. Modifying the α value at test time in our model has a similar effect as training the model with different λ_{cls} values.

H Face alignment

For the CelebA dataset we use the aligned version provided by the authors, which uses two landmark locations located at the eyes of each subject. Each image is first center-cropped to 178×178, and then resized to 128×128. For the in-the-wild high resolution images from Flickr, an internal face landmark detection network is used to automatically align and resize images to the mean CelebA face at 128×128. The location of the face landmarks used by the network are shown in Fig. 19. For the Cub200 dataset the face alignment to 128 × 128 uses four landmark locations: the beak, the crown, the forehead and the right eye. If the right eye is not visible, the image is left-right flipped.

Figure 19: An example of the locations of the 49 face landmarks used for the internal face landmark detection network.

I Network architectures and training details

The networks were trained on CelebA for 20 epochs and on Cub200 for 1545 epochs (due to the reduced size of this dataset). The Adam optimizer [18] is used with a learning rate of 0.0001, with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

Our network architectures are based on the StarGAN model. In the generator all transpose convolutions are replaced with bilinear resizing followed by convolution, and instance normalization is replaced by batch normalization. For the discriminator the StarGAN architecture is used without any modifications. In both tables the following notation is used, N is the number of output channels, K is the kernel size, S is the stride size, P is the padding size, and BN is batch normalization. The warping function, T, is implemented with a TensorFlow function during training, and with an OpenCV one for inference:
$T(\mathbf{x}, \mathbf{w}) =$ `tf.contrib.image.dense_image_warp(x, w)`,
$T(\mathbf{x}, \mathbf{w}) =$ `cv2.remap(x, w, interpolation=cv2.INTER_CUBIC)`.

Part	Input $\rightarrow$ Output Shape	Layer information
Down-sampling	$(h, w, 3 + r) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), ReLU, BN
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), ReLU, BN
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), ReLU, BN
Bottleneck	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 256)$	Bilinear resize
	$(\frac{h}{2}, \frac{w}{2}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S1, P1), ReLU, BN
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 128)$	Bilinear resize
	$(h, w, 64) \rightarrow (h, w, 64)$	CONV-(N64, K4x4, S1, P1), ReLU, BN
	$(h, w, 64) \rightarrow (h, w, 2)$	CONV-(N2, K7x7, S1, P1)

Table 2: Architecture for the warping network, W, the last layer is the displacement field $\mathbf{w}$, h and w denote the dimensionality of the input image, and r the number of attributes.

Part	Input $\rightarrow$ Output Shape	Layer information
Down-sampling	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(N1024, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{32}, \frac{w}{32}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(N2048, K4x4, S2, P1), Leaky ReLU
Output layer D	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1, K3x3, S1, P1)
Output layer C	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (1, 1, r)$	CONV-(N(r), K$\frac{h}{64}$x$\frac{w}{64}$, S1, P0)

Table 3: Architecture for the discriminator and the classifier networks, D and C. The kernel weights in the down-sampling layers are shared by D and C.

J Quantitative results: details

J.1 Accuracy vs identity preservation

In this section we give additional detail about the face re-identification network. We also provide attribute accuracy values and identity scores per attribute for the models used in the paper, namely, for StarGAN and StarGAN+ trained with $\lambda_{\text{cls}} = 0.25$ and for WarpGAN+ with $\lambda_{\text{cls}} = 2.00$.

J.1.1 Re-identification network

For the face re-identificaiton scores, presented in Fig. 8 in the paper, we use a Facenet model pretrained on the MS-Celeb-1M dataset [**?**]. This dataset consists of 10 million images and 100k unique identities. As both CelebA and MS-Celeb-1M were collected from publicly available Internet images, we expect some overlap between both datasets. This pretrained model is provided by the authors and is publicly available at https://github.com/davidsandberg/facenet.

Model	Smiling	Big nose	Arched eyebrows	Narrowed eyes	Pointy nose	Mean
StarGAN	0.65	0.60	0.64	0.66	0.68	0.65
StarGAN+	0.72	0.66	0.67	0.78	0.69	0.70
WarpGAN+	**0.83**	**0.73**	**0.81**	**0.87**	**0.82**	**0.81**
Real	1.00	1.00	1.00	1.00	1.00	1.00

Table 4: Quantitative comparison of the re-identification score on real and generated images on the CelebA dataset evaluated with the face re-identification network, higher is better.

J.1.2 Attribute classification accuracy

Model	Smiling	Big nose	Arched eyebrows	Narrowed eyes	Pointy nose	Mean
StarGAN	0.84	0.60	0.69	0.65	0.62	0.68
StarGAN+	**0.92**	**0.73**	**0.87**	**0.75**	**0.82**	**0.82**
WarpGAN+	0.72	0.72	0.83	0.74	0.74	0.75
Real	0.92	0.81	0.82	0.88	0.72	0.83

Table 5: Quantitative comparison of the attribute classification accuracy on real and generated images on the CelebA dataset evaluated with a separate classification network, higher is better.

J.2 User study

In the user study, for both experiments, to evaluate the reliability of the workers, a number of easy to classify images were mixed with the data, and used as a control. Workers needed to give the right answer to at least 90% of the control images for their data to be considered reliable. Images with fewer than 3 annotations are discarded, as they are

considered unreliable data. Finally, a simple majority voting scheme was used to determine the classification of each image.

For the experiment evaluating realism, typical failure cases for all models were shown to the workers before commencing the task, as examples of fake images. For the evaluation of the presence of the target attribute, to guide the workers, curated examples from training data edited with our model were shown to highlight the differences between the attributes.

Some images in the CelebA dataset contain border artifacts due to the alignment process that the authors used for the aligned version of the dataset. In order to get more reliable results, none of these images were included in the pool of 250 images used for the study.

J.2.1 Attribute classification accuracy

Model	Smiling	Big nose	Arched eyebrows	Narrowed eyes	Pointy nose	Mean
StarGAN	**0.85**	0.84	0.75	0.83	0.76	0.81
StarGAN+	**0.85**	0.84	**0.89**	0.86	0.83	**0.86**
WarpGAN+	0.63	**0.92**	0.83	**0.89**	**0.88**	0.84
Real	0.88	0.64	0.74	0.56	0.36	0.63

Table 6: Quantitative comparison of the attribute classification accuracy on real and generated images on the CelebA dataset evaluated with a user study, higher is better.

J.2.2 Realism accuracy

Model	Smiling	Big nose	Arched eyebrows	Narrowed eyes	Pointy nose	Mean
StarGAN	0.40	0.52	0.62	0.41	0.74	0.52
StarGAN+	0.37	0.40	0.59	0.38	0.60	0.46
WarpGAN+	**0.42**	**0.64**	**0.79**	**0.57**	**0.82**	**0.65**
Real	0.97	0.89	0.98	0.96	0.94	0.95

Table 7: Quantitative comparison of image realism both on real and generated images on the CelebA dataset evaluated with a user study, higher is better.